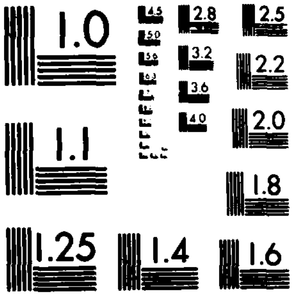
AD-A153 843 PROGRAM DISAS - A COMPUTER PROGRAM TO OBTAIN HARD-COPY 1/1
PLOTS OF AN IMAGE..(U) AERONAUTICAL RESEARCH LABS
MELBOURNE (AUSTRALIA) R BATEMAN JAN 85 ARL/SYS-TM-76

UNCLASSIFIED F/G 9/2 NL

END
DATE
FILMED
6 85



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ARL-SYS-TM-76 AR-003-989

AUSTRALIA

# DEPARTMENT OF DEFENCE

## DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

## AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Systems Technical Memorandum 76

# PROGRAM DISAS - A COMPUTER PROGRAM TO OBTAIN HARD-COPY PLOTS OF AN IMAGE DISPLAYED ON A VECTOR GRAPHICS DEVICE IN A LOCAL-AREA-NETWORK

by

R. BATEMAN

Approved for Public Release

COPY No JANUARY 1985

AR-003-989

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

Systems Technical Memorandum 76

# PROGRAM DIAS - A COMPUTER PROGRAM TO OBTAIN HARD-COPY PLOTS OF AN IMAGE DISPLAYED ON A VECTOR GRAPHICS DEVICE IN A LOCAL-AREA-NETWORK

by

R. BATEMAN

## SUMMARY

Program DISAS is a program which supports the computer graphics facilities of Combat Effectiveness Group at ARL. It operates on a local-area-network incorporating a network of PDP11 minicomputers, a Sanders Graphic-7 vector graphics device and a Versatec electrostatic printer/plotter.

Program DISAS produces hard-copy plots on the Versatec printer/plotter of a selected image displayed on the screen of the SANDERS Graphic-7 vector graphics device.

A listing of the program can be obtained from the Principal Officer of Combat Effectiveness Group.

POSTAL ADDRESS: Director, Aeronautical Research Laboratories,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

Contents

Page No.

## 1. INTRODUCTION

The program DISAS is one of the programs written to support the computer graphics facilities of Combat Effectiveness Group (C.E.G) at the Aeronautical Research Laboratories.

The system upon which this program operates is a local-area-network running under the STAR-11 operating system. It incorporates a PDP11/35 minicomputer as host and a number of micro-computers as satellites. The satellite that is used by C.E.G is a PDP11/23 running under the RT-11 operating system. It in turn acts as host to a SANDERS Graphic-7 vector graphics device whose processor emulates the instruction set of a PDP11/34 minicomputer. Reference 1 describes the function of the Graphic-7 processor including its programming, data communication and image generation. Reference 2 describes the use of the graphic-7 coordinate converter. The supported hard-copy facility is a Versatec electrostatic printer/plotter Model V80 which runs under control of the PDP11/35 host computer in stand-alone mode.

The software package which controls the display of images on the Graphic-7 screen is called Fortran Support Package(FSP)* and resides on the PDP/11 microcomputer. The purpose of this package is to convert the image processing instructions from the user's program into graphics instructions which are recognized by the Graphic-7 graphics processor. It then prepares these instructions for transmission to the graphics processor. Reference 3 describes the use of the Fortran Support Package.

The Versatec V80 plotting software package is called Versaplot + and also exists on the PDP11/23. Reference 4 is the user's manual for this software package.

The purpose of the program DISAS is to obtain a hard-copy plot on the Versatec printer/plotter of the image displayed on the Graphic-7 screen. The set of Graphic-7 instructions that generated the displayed image from which the hard-copy plot is to be developed must have been dumped from Graphic-7 memory. This is the input file to the program DISAS and exists on a PDP11/23 disk file with a user-selected name.

* Fortran Support Package (FSP) is proprietary to Sanders Associates Inc.

+ Versaplot is proprietary to VERSATEC Inc.

## 2. Program DISAS User's Guide

The program DISAS is comprised of the two source modules DISAS1.FOR and DISAS2.MAC. DISAS1.FOR includes the bulk of the program and is written in Fortran while DISAS2.MAC includes the subroutine to identify an instruction word and is written in the assembly language MACRO-11.

To produce the executable file DISAS.EXE follow the compile and link procedure shown in Fig 2.1. Then to run the program type to the monitor prompt RUN DISAS.EXE

The input file to this program consists of machine-code instructions which have been dumped to the PDP11/23 from the Graphic-7 memory. One source of instigation of this dump is from the selection of the appropriate option to the main menu in program COMBAT. For a complete discussion on this aspect see Reference 5. Section 4.1 gives a description of the format of this input data file.

When DISAS is run, the user is prompted for the filename of the input file. This is the only user-interaction required during one run of DISAS.

The assembly language program PASS1.DAT and the Fortran program PASS2.DAT are produced by DISAS. PASS1.DAT is produced by the first phase of the program and is used as input to the second phase which produces the file PASS2.DAT. PASS2.DAT is a Fortran program which incorporates the Versatec plotting commands to produce the hard-copy plot of the required image. It is compiled, linked and run as shown in Fig 2.2.

The files produced by PASS2 are VECTR1.BIN and PARM.BIN. These files are used as input to the program RASM which is the plotting control program supplied with the Versaplot software package. An example of the process to obtain a hard-copy plot using program RASM is shown in Fig 2.3, and an example of the output produced by this process is shown in Fig 2.4.

```
.FORTRAN DISAS1
.MACRO DISAS2
.LINK/EXE:DISAS.EXE DISAS1,DISAS2
```

Fig 2.1 The compile and link procedure to build the executable file DISAS.EXF

```
.ASS LA: VPO:
.ASS LA: LP:
.ASS LA: 6
.FORT PASS2.DAT
.LINK/EXE:VPO VPO(PASS2,MAPPED,PEPLIB)
.RUN PASS2
```

- where LA is the logical area of the PDP11/23 disk where the appropriate files exist.

Fig 2.2 The compile,link and run sequence of the program PASS2.DAT

Step 1. Copy the plot files, VECTR1.BIN and PARM.BIN, to a floppy disk which has the plotting control program RASM on it by following the sequence:

```
.COPY PARM.BIN,VECTR1.BIN DLO:
. <BREAK>
@173000G

.COPY PARM.BIN,VECTR1.BIN FDD:
```

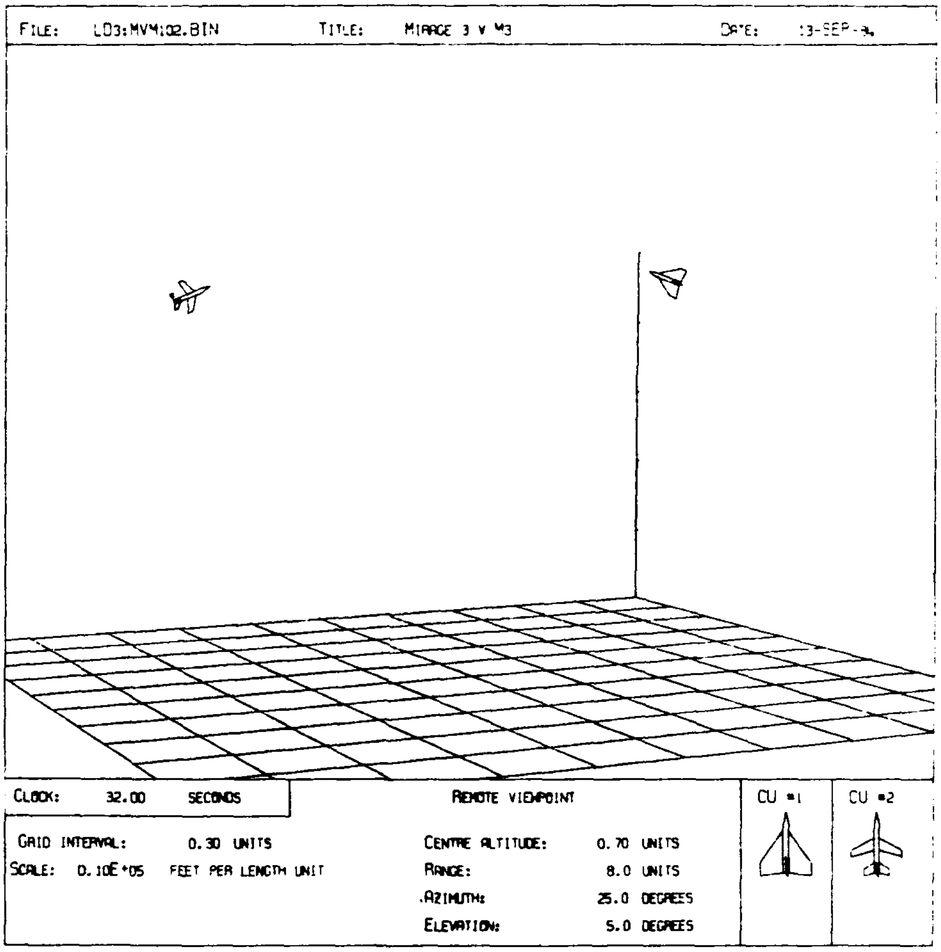
- where FDD is the floppy disk drive chosen

Step 2. Unload the foreground of the STAR-11 network by following the sequence:

```
.EXIT
^F
^C
^C
^B
UNLOAD F
.ASS FDD: VPO:
.RUN VPO:RASM
```

- where the "^" symbol represents the control key of the keyboard. The required sequence is obtained by holding the control key down while pressing the associated key

Fig 2.3 The sequence to produce a hard copy plot once the plot files are produced on the PDP11/23.



FILE: LD3:MVM102.BIN  TITLE: MIRAGE 3 V M3  DATE: 13-SEP-84

CLOCK: 32.00 SECONDS

REMOTE VIEWPOINT

GRID INTERVAL: 0.30 UNITS

SCALE: 0.10E+05 FEET PER LENGTH UNIT

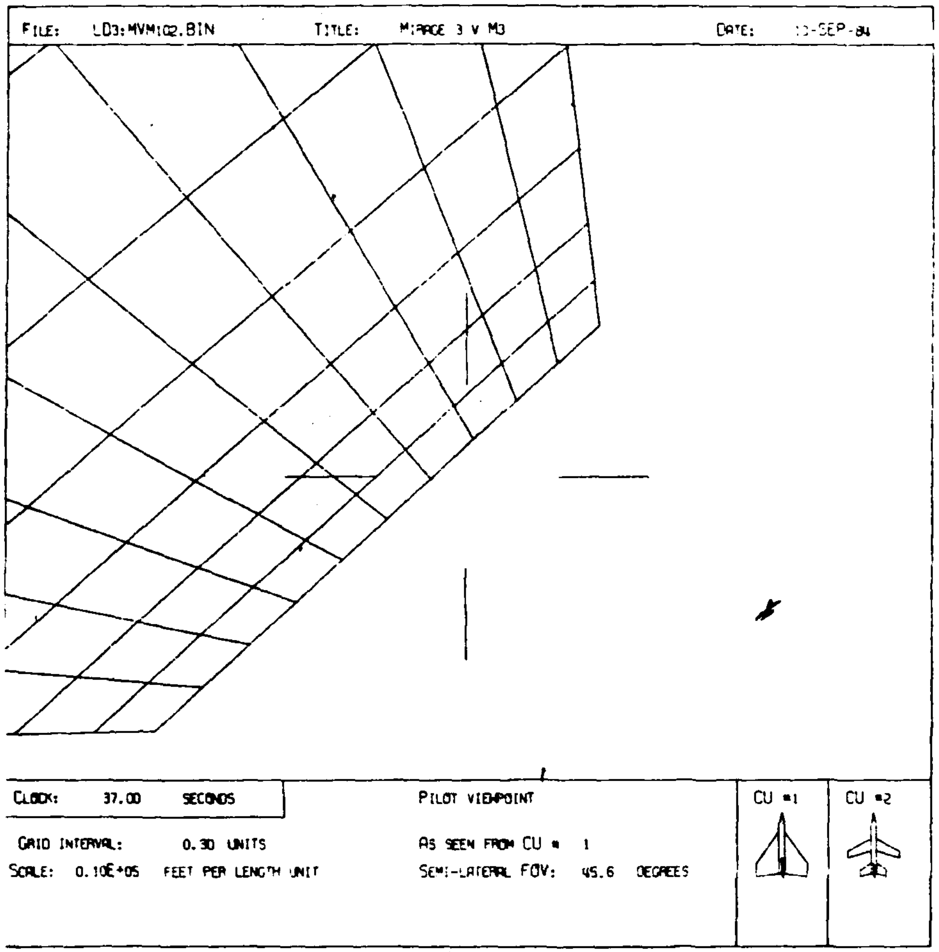
CENTRE ALTITUDE: 0.70 UNITS

RANGE: 8.0 UNITS

AZIMUTH: 25.0 DEGREES

ELEVATION: 5.0 DEGREES

Fig. 2.4(a) Example output by DISAS



FILE: LD3:MVM102.BIN TITLE: MIRAGE 3 V M3 DATE: [illegible]-SEP-84

CLOCK: 37.00 SECONDS

PILOT VIEWPOINT

GRID INTERVAL: 0.30 UNITS

SCALE: 0.10E+05 FEET PER LENGTH UNIT

AS SEEN FROM CU # 1

SEMI-LATERAL FOV: 45.6 DEGREES

Fig. 2.4(b) Example output by DISAS

## 3. PROGRAM DESCRIPTION and DESIGN

There are two phases to the program DISAS. The first phase produces an assembly language program from the input file of Graphic-7 machine code instructions and produces the file named PASS1.DAT. The second phase uses PASS1.DAT as input and translates the assembly instructions into a Fortran program which utilizes the Versatec plotting primitives. Fig. 3.1 shows an hierarchical structure chart of the program.

There are nineteen Graphic-7 control and display instructions accounted for in program DISAS. Fig 3.2 lists and describes the format of these instructions.

### 3.1 Phase 1

The user enters the filename of the file of Graphic-7 machine instructions in reply to a prompt. Section 4.1 describes the format of this file. The program reads through the data file and produces an array of Graphic-7 source picture numbers and their start addresses. When this is completed the data file is rewound back to its start. Fig 3.3 shows this process.

The input file is then re-read, one line of 10 words at a time until a block of 100 words is read or the end-of-picture is detected. Each word of this block is identified and its op-code and argument/s are dissassembled and output to the file PASS1.DAT as appropriate. The start of each picture is recognized and a label is written to the file PASS1.DAT. If the word was recognized as a CALL SUBROUTINE instruction then the next word is the start address of the destination picture. A search is made through the array of picture numbers and start addresses for a match for this address. The associated picture number is the destination picture. Fig 3.4 gives a description of this sequence.

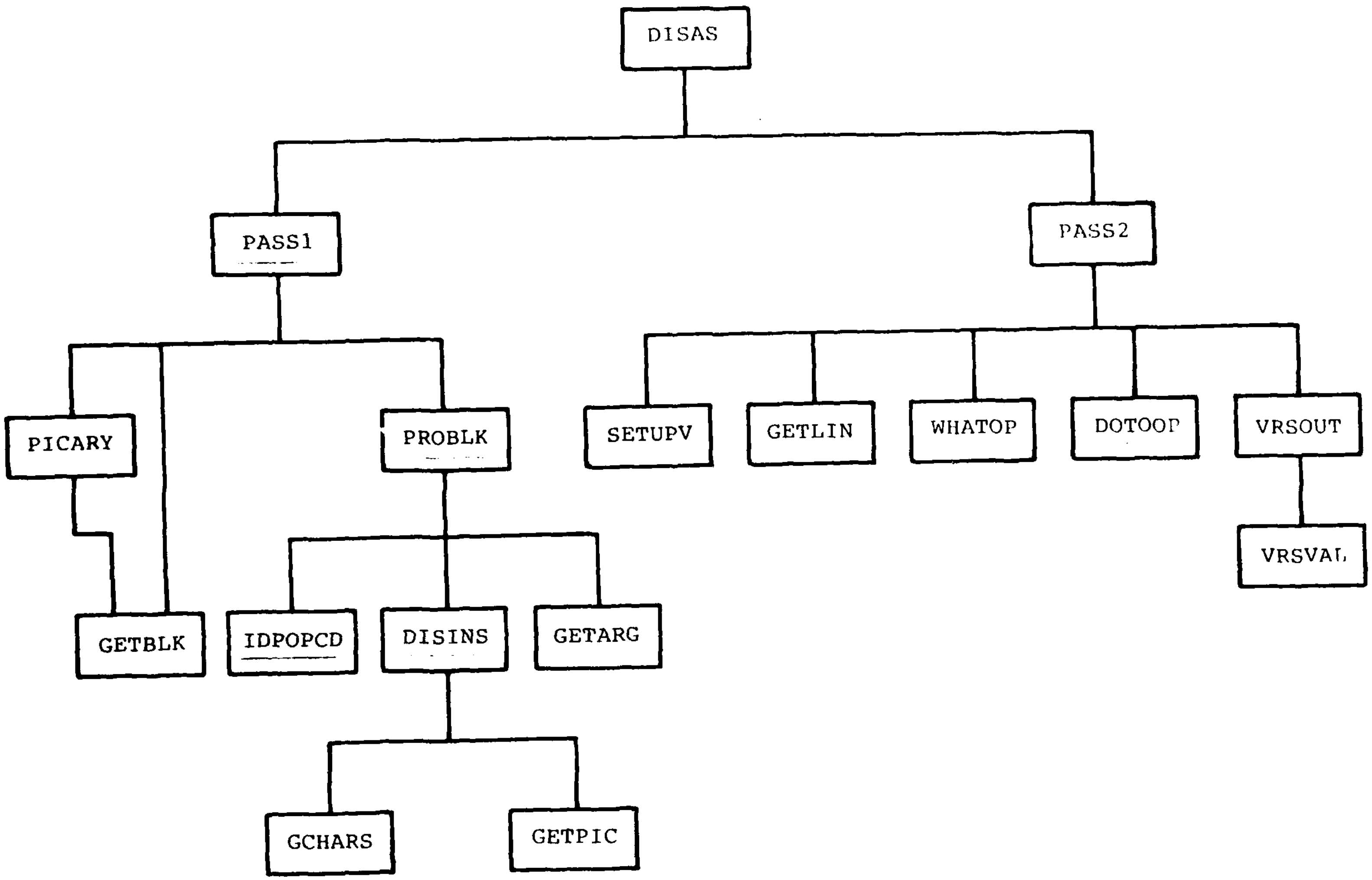


Fig. 3.1 Hierarchical structure chart of DISAS

| | | |
|---|---|---|
| CALL | 0 000 010 001 x x x x x x | Call Subroutine |
| | Subroutine Address | |
| RTRN | 0 000 010 011 x x x x x x | Return |
| JMPR/NOOP | 0 000 101 ± Jump ammount | Jump Relative/ No Op |
| LDDZ | 0 001 0 11 Bits Data | Load Z-axis Register |
| LDDP | 0 001 1 11 Bits Data | Load Display Param. Register |
| LDXA | 0 010 0 ± X coordinate | Load X absolute |
| LDXR | 0 010 1 ± X increment | Load DX relative |
| DRXA | 0 011 0 ± X coordinate | Draw X absolute |
| DRXR | 0 011 1 ± X increment | Draw DX relative |
| DRYA | 0 100 0 ± Y coordinate | Draw Y absolute |
| DRYR | 0 100 1 ± Y increment | Draw DY relative |
| MVXA | 0 101 0 ± X coordinate | Move X absolute |
| MVXR | 0 101 1 ± X increment | Move DX relative |
| MVYA | 0 110 0 ± Y coordinate | Move Y absolute |
| MVYR | 0 110 1 ± Y increment | Move DY relative |
| PPLR | 1 1 ±5 Bits Y 0 0 ±5 Bits X | Point Plot relative |
| LDTI | 1 1 00 000 001 Increment | Load Text Increment register |
| TEXT | 1 2nd ASCII 1 1st ASCII | Draw two text characters |
| CHAR | 1 001 11B1 1 ASCII char | Draw single text character |

Fig. 3.2 Graphic-7 control and display instructions incorporated in DISAS

```
DO UNTIL END-OF-FILE
        READ PICTURE NUMBER AND ITS START ADDRESS
        LOAD THESE VALUES INTO ARRAY

        DO UNTIL END-OF-PICTURE
                SKIP OVER LINES OF DATA
        END DO

END DO
REWIND FILE BACK TO START
```

Fig 3.3 Pseudo code of operation to build up array of picture numbers and start addresses.

```
DO UNTIL END-OF-FILE
     READ picture number and Write this label to PASS1.DAT

     DO UNTIL END-OF-PICTURE

        DO UNTIL 100 words read in or END-OF-PICTURE
             READ line of 10 words and append to ARRAY(I)
             SEARCH through these 10 words for RETURN instruction
             IF FOUND THEN END-OF-PICTURE
        END DO

        DO FOR ALL words in this block
             IDENTIFY its op-code
             DISSASSEMBLE arguments as appropriate
             OUTPUT dissassembled instruction to PASS1.DAT
        END DO

     END DO

END DO

CLOSE INPUT FILE
```

Fig 3.4 Pseudo code for dissassembler process of phase 1

## 3.2 Phase 2

The file of assembled instructions, PASS1.DAT is reopened as READONLY.

The first action of Phase 2 is to write Fortran code to the MAIN section of PASS2.DAT which will initialize the Versatec plotter and enable an eight inch box to be drawn on the hard copy plot to represent the Graphic-7 screen. Then Fortran code to call Picture 1 is written to the MAIN section because all Graphic-7 control by FSP is determined from Picture 1.

Three small Fortran subroutines are written to PASS2.DAT to enable relative move and draw instructions and symbols to be plotted. These are called VRELM, VRELD and VSYMB respectively. See Fig 3.5 for a listing of these subroutines.

This second phase reads each line of the file in turn, identifying the op-code and isolating the argument/s as appropriate. Some instructions require that the succeeding line be processed in order to complete the dissassembly process. The assembly instruction is dissassembled into an equivalent Fortran-compatible Versatec plotting command and then output to the file PASS2.DAT. PASS2.DAT is in the format of a Fortran program. Fig 3.6 shows the sequence of phase 2 of the dissassembler process.

```
SUBROUTINE VRELM(IXREL,IYREL)
CALL WHERE(XNOW,YNOW,DFACT)
TOX=XNOW+ ((FLOAT(IXREL)/1023.0)*8.0)
TOY=YNOW+ ((FLOAT(IYREL)/1023.0)*8.0)
CALL PLOT(TOX,TOY,3)
RETURN
END

SUBROUTINE VRELD(IXREL,IYREL)
CALL WHERE(XNOW,YNOW,DFACT)
TOX=XNOW+ ((FLOAT(IXREL)/1023.0)*8.0)
TOY=YNOW+ ((FLOAT(IYREL)/1023.0)*8.0)
CALL PLOT(TOX,TOY,2)
RETURN
END
```

Fig 3.5 a. Listing of the subroutines VRELM and VRELD

```
SUBROUTINE VSYMB(HT,ITEXT,ROT,NC)
CALL WHERE(XPOS,YPOS,DFACT)
CALL SYMBOL(XPOS,YPOS,HT,ITEXT,ROT,NC)
IF(ROT.GT.80.0)YPOS=YPOS+(HT*0.8)
IF(ROT.LT.10.0)XPOS=XPOS+(HT*0.8)
CALL PLOT(XPOS,YPOS,3)
RETURN
END
```

Fig 3.5 b. Listing of the subroutine VSYMB

```
WRITE Fortran code to draw box
WRITE Fortran code to initialize Versatec plotter
WRITE Fortran subroutines to enable move and draw instructions
      and symbol plotting

DO UNTIL END-OF-FILE

    DO UNTIL complete Versatec comand built up
        READ line of assembly code from PASS1.DAT
        IDENTIFY this op-code
        EXTRACT argument/s as appropriate
        BUILD UP Versatec plot command
    END DO

    WRITE Versatec plot command to PASS2.DAT

END DO
```

Fig 3.6 Pseudocode describing the second phase of the dissassembler

## 4. FILE FORMATS

There are two files produced by DISAS and one file required as input. The produced files are called PASS1.DAT and PASS2.DAT while the input file name is user defined.

### 4.1 Input File

This file consists of the dump of the instructions from the Graphic-7 memory which generated the image seen on the Graphic-7 screen at the selected time.

The file is organized into segments where each segment represents one Fortran Support Package picture. The picture number and its start address are included at the start of the segment as a title line.

The data within each segment is in the format of 10 octal words per line with the RTRN instruction (octal 2300) being the final word in the segment. Fig 4.1 is an example of the input file.

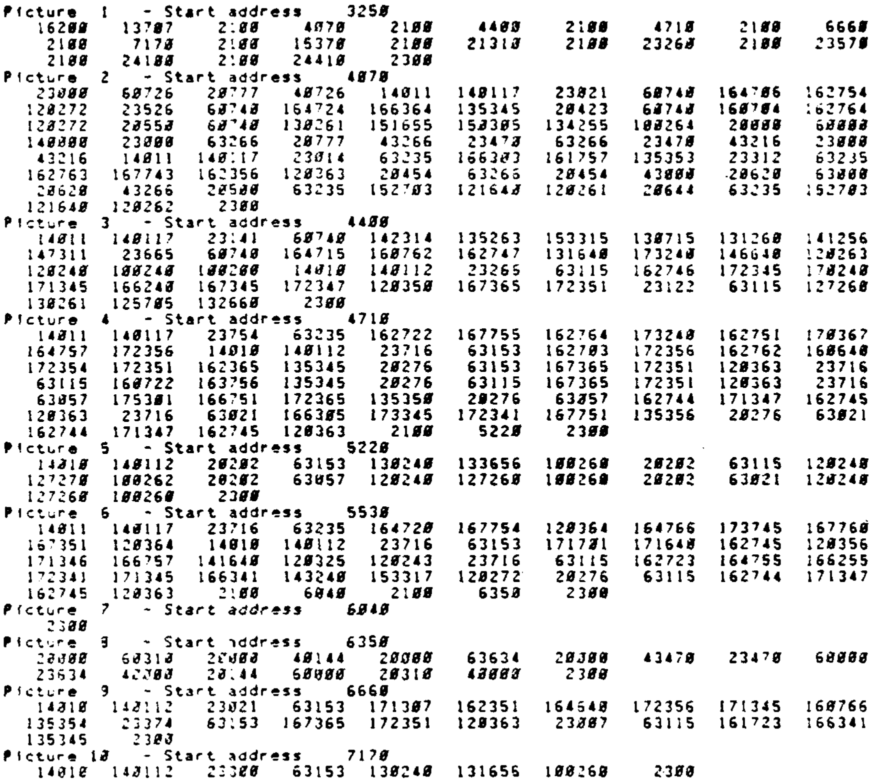
### 4.2 PASS1.DAT

PASS1.DAT is the file produced by the first phase of DISAS. It is in the format of a Graphic-7 assembly language program. it has labels representing picture or subroutine start locations and instructions consisting of a four-character op-code and octal argument/s as applicable. Fig 4.2 shows an example of such a file.

### 4.3 PASS2.DAT

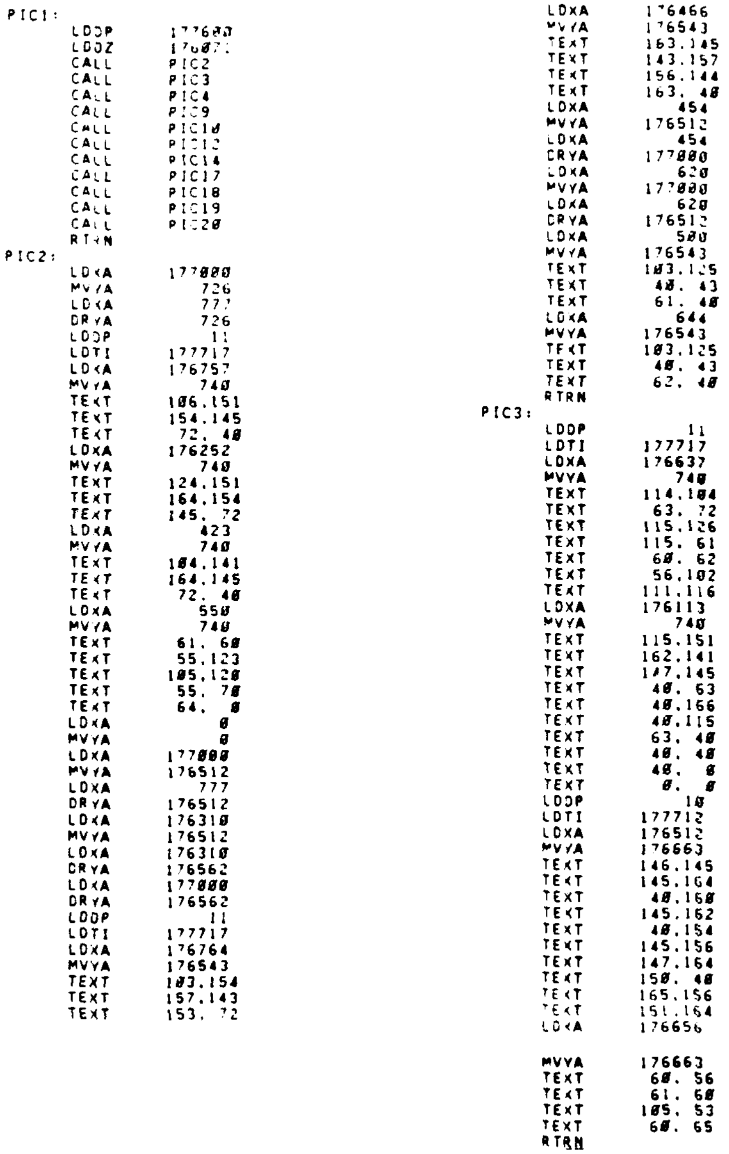
This is the resultant Fortran program which will generate the hard-copy plots on the Versatec plotter. It contains the subroutines VRELM and VRELD which allow the utilizaton of relative move and draw instructions and the subroutine VSYMB to allow character symbols to be drawn on the Versatec plotter.

The Graphic-7 instructions are now represented by Fortran statements utilizing the equivalent Versaplot plotting primitives. Fig 4.3 shows an example segment of the file PASS2.DAT.



```
Picture  1   - Start address     3250
   16200    13707     2100     4070     2100     4400     2100     4710     2100     6660
    2100     7170     2100    15370     2100    21310     2100    23260     2100    23570
    2100    24100     2100    24410     2300
Picture  2   - Start address     4070
   23000    60726    20777    40726    14011   140117    23021    60740   164706   162754
  120272    23526    60740   164724   166364   135345    20423    60740   160704   162764
  120272    20550    60740   130261   151655   150305   134255   100264    20000    60000
  140000    23000    63266    20777    43266    23470    63266    23470    43216    23000
   43216    14011   140117    23014    63235   166303   161757   135353    23312    63235
  162763   167743   162356   120363    20454    63266    20454    43000    20620    63000
   20620    43266    20500    63235   152703   121640   120261    20644    63235   152703
  121640   120262     2300
Picture  3   - Start address     4400
   14011   140117    23141    60740   142314   135263   153315   130715   131260   141256
  147311    23665    60740   164715   160762   162747   131640   173240   146640   120263
  120240   100240   100200    14010   140112    23265    63115   162746   172345   170240
  171345   166240   167345   172347   120350   167365   172351    23122    63115   127260
  130261   125705   132660     2300
Picture  4   - Start address     4710
   14011   140117    23754    63235   162722   167755   162764   173240   162751   170367
  164757   172356    14010   140112    23716    63153   162703   172356   162762   160640
  172354   172351   162365   135345    20276    63153   167365   172351   120363    23716
   63115   160722   163756   135345    20276    63115   167365   172351   120363    23716
   63057   175301   166751   172365   135350    20276    63057   162744   171347   162745
  120363    23716    63021   166305   173345   172341   167751   135356    20276    63021
  162744   171347   162745   120363     2100     5220     2300
Picture  5   - Start address     5220
   14010   140112    20202    63153   130240   133656   100260    20202    63115   120240
  127270   100262    20202    63057   120240   127260   100260    20202    63021   120240
  127260   100260     2300
Picture  6   - Start address     5530
   14011   140117    23716    63235   164720   167754   120364   164766   173745   167760
  167351   120364    14010   140112    23716    63153   171701   171640   162745   120356
  171346   166757   141640   120325   120243    23716    63115   162723   164755   166255
  172341   171345   166341   143240   153317   120272    20276    63115   162744   171347
  162745   120363     2100     6040     2100     6350     2300
Picture  7   - Start address     6040
    2300
Picture  8   - Start address     6350
   20000    60310    20000    40144    20000    63634    20300    43470    23470    60000
   23634    42000    20144    60000    20310    40000     2300
Picture  9   - Start address     6660
   14010   140112    23021    63153   171307   162351   164540   172356   171345   160766
  135354    23374    63153   167365   172351   120363    23007    63115   161723   166341
  135345     2300
Picture 10   - Start address     7170
   14010   140112    23300    63153   130240   131655   100260     2300
```

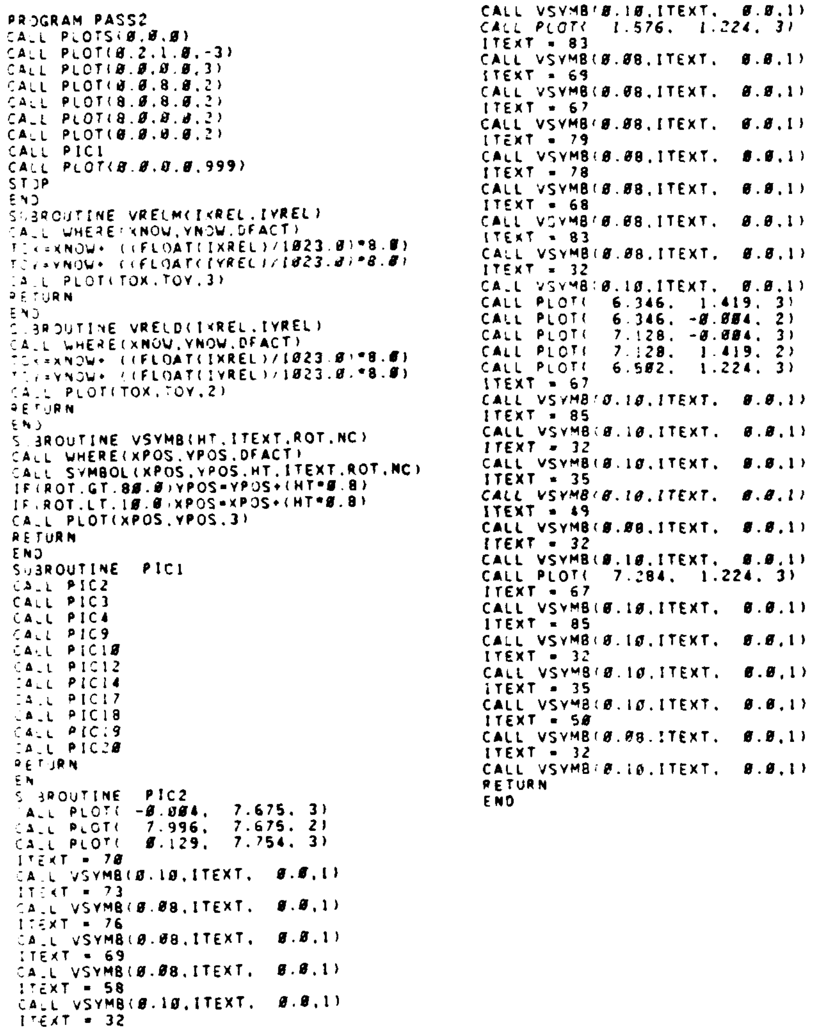
Fig. 4.1  Example of the input file to DISAS



```
PIC1:
        LDOP    1776ØØ
        LDOZ    176Ø71
        CALL    PIC2
        CALL    PIC3
        CALL    PIC4
        CALL    PIC9
        CALL    PIC1Ø
        CALL    PIC12
        CALL    PIC14
        CALL    PIC17
        CALL    PIC18
        CALL    PIC19
        CALL    PIC2Ø
        RTRN
PIC2:
        LDXA    177ØØØ
        MVYA       726
        LDXA       777
        DRYA       726
        LDOP        11
        LDTI    177717
        LDXA    176757
        MVYA       74Ø
        TEXT    1Ø6.151
        TEXT    154.145
        TEXT     72. 4Ø
        LDXA    176252
        MVYA       74Ø
        TEXT    124.151
        TEXT    164.154
        TEXT    145. 72
        LDXA       423
        MVYA       74Ø
        TEXT    1Ø4.141
        TEXT    164.145
        TEXT     72. 4Ø
        LDXA       55Ø
        MVYA       74Ø
        TEXT     61. 6Ø
        TEXT     55.123
        TEXT    1Ø5.12Ø
        TEXT     55. 7Ø
        TEXT     64.  Ø
        LDXA         Ø
        MVYA         Ø
        LDXA    177ØØØ
        MVYA    176512
        LDXA       777
        DRYA    176512
        LDXA    17631Ø
        MVYA    176512
        LDXA    17631Ø
        DRYA    176562
        LDXA    177ØØØ
        DRYA    176562
        LDOP        11
        LDTI    177717
        LDXA    176764
        MVYA    176543
        TEXT    1Ø3.154
        TEXT    157.143
        TEXT    153. 72
        LDXA    176466
        MVYA    176543
        TEXT    163.145
        TEXT    143.157
        TEXT    156.144
        TEXT    163. 4Ø
        LDXA       454
        MVYA    176512
        LDXA       454
        DRYA    177ØØØ
        LDXA       62Ø
        MVYA    177ØØØ
        LDXA       62Ø
        DRYA    176512
        LDXA       5ØØ
        MVYA    176543
        TEXT    1Ø3.125
        TEXT     4Ø. 43
        TEXT     61. 4Ø
        LDXA       644
        MVYA    176543
        TEXT    1Ø3.125
        TEXT     4Ø. 43
        TEXT     62. 4Ø
        RTRN
PIC3:
        LDOP        11
        LDTI    177717
        LDXA    176637
        MVYA       74Ø
        TEXT    114.1Ø4
        TEXT     63. 72
        TEXT    115.126
        TEXT    115. 61
        TEXT     6Ø. 62
        TEXT     56.1Ø2
        TEXT    111.116
        LDXA    176113
        MVYA       74Ø
        TEXT    115.151
        TEXT    162.141
        TEXT    147.145
        TEXT     4Ø. 63
        TEXT     4Ø.166
        TEXT     4Ø.115
        TEXT     63. 4Ø
        TEXT     4Ø. 4Ø
        TEXT     4Ø.  Ø
        TEXT      Ø.  Ø
        LDOP        1Ø
        LDTI    177712
        LDXA    176512
        MVYA    176663
        TEXT    146.145
        TEXT    145.164
        TEXT     4Ø.16Ø
        TEXT    145.162
        TEXT     4Ø.154
        TEXT    145.156
        TEXT    147.164
        TEXT    15Ø. 4Ø
        TEXT    165.156
        TEXT    151.164
        LDXA    176656

        MVYA    176663
        TEXT     6Ø. 56
        TEXT     61. 6Ø
        TEXT    1Ø5. 53
        TEXT     6Ø. 65
        RTRN
```

Fig. 4.2 Example of the file PASS1.DAT



```
PROGRAM PASS2
CALL PLOTS(Ø.Ø,Ø)
CALL PLOT(Ø.2,1.Ø,-3)
CALL PLOT(Ø.Ø,Ø.Ø,3)
CALL PLOT(Ø.Ø,8.Ø,2)
CALL PLOT(8.Ø,8.Ø,2)
CALL PLOT(8.Ø,Ø.Ø,2)
CALL PLOT(Ø.Ø,Ø.Ø,2)
CALL PIC1
CALL PLOT(Ø.Ø,Ø.Ø,999)
STOP
END
SUBROUTINE VRELM(IXREL,IYREL)
CALL WHERE(XNOW,YNOW,DFACT)
TOX=XNOW+ ((FLOAT(IXREL)/1Ø23.Ø)*8.Ø)
TOY=YNOW+ ((FLOAT(IYREL)/1Ø23.Ø)*8.Ø)
CALL PLOT(TOX,TOY,3)
RETURN
END
SUBROUTINE VRELD(IXREL,IYREL)
CALL WHERE(XNOW,YNOW,DFACT)
TOX=XNOW+ ((FLOAT(IXREL)/1Ø23.Ø)*8.Ø)
TOY=YNOW+ ((FLOAT(IYREL)/1Ø23.Ø)*8.Ø)
CALL PLOT(TOX,TOY,2)
RETURN
END
SUBROUTINE VSYMB(HT,ITEXT,ROT,NC)
CALL WHERE(XPOS,YPOS,DFACT)
CALL SYMBOL(XPOS,YPOS,HT,ITEXT,ROT,NC)
IF(ROT.GT.8Ø.Ø)YPOS=YPOS+(HT*Ø.8)
IF(ROT.LT.1Ø.Ø)XPOS=XPOS+(HT*Ø.8)
CALL PLOT(XPOS,YPOS,3)
RETURN
END
SUBROUTINE  PIC1
CALL PIC2
CALL PIC3
CALL PIC4
CALL PIC9
CALL PIC1Ø
CALL PIC12
CALL PIC14
CALL PIC17
CALL PIC18
CALL PIC19
CALL PIC2Ø
RETURN
EN
SUBROUTINE  PIC2
CALL PLOT( -Ø.ØØ4,  7.675, 3)
CALL PLOT(  7.996,  7.675, 2)
CALL PLOT(  Ø.129,  7.754, 3)
ITEXT = 7Ø
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 73
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 76
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 69
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 58
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
CALL PLOT(  1.576,  1.224, 3)
ITEXT = 83
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 69
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 67
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 79
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 78
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 68
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 83
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
CALL PLOT(  6.346,  1.419, 3)
CALL PLOT(  6.346, -Ø.ØØ4, 2)
CALL PLOT(  7.128, -Ø.ØØ4, 3)
CALL PLOT(  7.128,  1.419, 2)
CALL PLOT(  6.5Ø2,  1.224, 3)
ITEXT = 67
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 85
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 35
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 49
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
CALL PLOT(  7.284,  1.224, 3)
ITEXT = 67
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 85
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 35
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
ITEXT = 5Ø
CALL VSYMB(Ø.Ø8,ITEXT,  Ø.Ø,1)
ITEXT = 32
CALL VSYMB(Ø.1Ø,ITEXT,  Ø.Ø,1)
RETURN
END
```

Fig. 4.3 Example of the file PASS2.DAT

## 5. SUBROUTINE DESCRIPTION

### 5.01 PROGRAM MAIN

Purpose: Prepare the input file and control the flow of execution of the program.

Method:

- Accept the filename of input Graphic-7 machine code instructions and open this file as READONLY
- Call subroutine PASS1 to control phase 1 of program
- Call subroutine PASS2 to control phase 2 of program
- Close all open files

SUBROUTINES CALLED:
PASS1,PASS2

CALLED BY:
Nil

### 5.02 SUBROUTINE PASS1

Purpose: Control execution of phase 1 of the program.

Method:

- Open the output file of phase 1 (PASS1.DAT)
- Call subroutine PICARY to build the array of picture numbers and start addresses
- Read the picture number from the input file and write the appropriate label to PASS1.DAT for each picture
- Call subroutine GETBLK to input a block of data words from the input file.
- Call subroutine PROBLK to process this block

SUBROUTINES CALLED:
PICARY,GETBLK,PROBLK

CALLED BY:
MAIN

## 5.03 SUBROUTINE PICARY

Purpose: Load picture numbers and start addresses into the array PICSAD.

METHOD:

- Read picture header line including picture number and start address

- Call subroutine GETBLK to skip over all of this picture.

- Repeat for all pictures in the input file

- Rewind the input file back to its start

SUBROUTINES CALLED:
GETBLK

CALLED BY:
PASS1

## 5.04 SUBROUTINE GETBLK(ARRAY,NWORDS,ENDPIC,EOF)

Purpose: Input a block of up to 100 data words from the input file.

Method:

- Read a line of 10 words from the input file.

- If the RTRN instruction, signifying the end of this picture, is found in this line then return to the calling subroutine with the data block read in

- If the RTRN instruction is not found in this line then repeat the process until either a block of 100 words read in or the RTRN instruction is encountered.

SUBROUTINES CALLED:
Nil

CALLED BY:
PASS1

## 5.05 SUBROUTINE PROBLK(ARRAY,NWORDS)

Purpose: Control the processing of the data block which was previously read. There are NWORDS in the current data block which is stored in the array ARRAY.

Method:

- Call subroutine IDOPCD to identify the op-code of the current instruction.
- Call subroutine GETARG to separate the argument/s of this instruction.
- Call subroutine DISINS to dissassemble and output this instruction.

SUBROUTINES CALLED:
IDOPCD,PROBLK,DISINS

CALLED BY:
PASS1

## 5.06 SUBROUTINE IDOPCD(IWRD,IOP)

Purpose: Identify the op-code of the current instruction. This subroutine is written in MACRO 11.

Method:

- Determine if the instruction is one of the op-codes which imply a specific value. These op-codes have no arguments in this data word and are CALL, RTRN and NOOP.
- If not then compare each instruction with the specified range of the remaining op-codes (accounting for the highest and the lowest possible values for its argument/s).
- Assign a value to the variable IOP to represent the matched op-code.
- If no match found then let IOP=0

SUBROUTINES CALLED:
Nil

CALLED BY:
PROBLK

## 5.07 SUBROUTINE GETARG(IWRD,IOP,ARG)

Purpose: Return the value of the argument/s, ARG, for the given instruction in IWRD which has op-code number IOP.

Method:

- If IOP equals 0,1,2,3 or 19 then ARG = 0 (there are no arguments)

- If IOP lies between 4 and 15 inclusive then the instruction is a display instruction. The argument is represented in 2's complement notation and bit 10 is a sign bit.
  ARG=IWRD - ((IWRD/"4000)*"4000)

| | |
|---|---|
| op code | 2's complement argument |

- If IOP = 16 then the word is an LDTI instruction .
  ARG=IWRD-((IWRD/"100)*"100)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | text increment |

- If IOP = 17 then the word is a TEXT instruction. The arguments are two ASCII characters and let the argument ARG equal the whole word.
  ARG=IWRD

| | | | |
|---|---|---|---|
| 1 | 2nd character | 1 | 1st character |

- If IOP = 18 then the word is a CHAR instruction. The argument is one ASCII character.
  ARG=IWRD - ((IWRD/"200)*"200)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 | B | 1 | 1 | character |

SUBROUTINES CALLED:
Nil

CALLED BY:
PROBLK

## 5.08 SUBROUTINE DISINS(ARRAY,IW,IWRD,IOP,ARG)

Purpose: Write the dissassembled instruction to the file PASS1.DAT in the format:

Op-code, Argument/s

Method:

- Set up a text array of op-codes indexed by IOP
- If IOP = 17 then the instruction is the TEXT op-code. Call subroutine GCHARS to separate the two ASCII characters from the instruction word.
- Write the op-code followed by the applicable arguments to the file PASS1.DAT

SUBROUTINES CALLED:
GCHARS

CALLED BY:
PROBLK

## 5.09 SUBROUTINE GCHARS(ARG,ICHAR)

Purpose: The current word is a TEXT instruction. This subroutine separates the two ASCII characters from the word ARG.

Method:

- Decode the integer word ARG into two bytes.
- Clear bit 7 of each of these two bytes by
  ```
  ICHAR(I)=ICHAR(I)-"200
  ```

SUBROUTINES CALLED:
Nil

CALLED BY:
DISINS

## 5.10 SUBROUTINE PASS2

Purpose: Control phase 2 of the program.

Method:

- Re-open PASS1.DAT as READONLY

- Open the output file PASS2.DAT

- Call subroutine SETUPV to write Fortran code to PASS2.DAT to setup Versatec plotter and enable relative move and draw primatives and symbol plotting.

- Input a line of instructions from PASS1.DAT

- Identify this line and its arguments

- Disassemble this instruction into Fortran code. If the line of Fortran code is complete then write it to PASS2.DAT. Else read the next line from PASS1.DAT and process it.

SUBROUTINES CALLED:
SETUPV,GETLIN,WHATOP,DOTOOP,VRSOUT

CALLED BY:
MAIN

## 5.11 SUBROUTINE SETUPV

Purpose: Write Fortran code to MAIN section of PASS2.DAT to enable an eight inch box to be drawn on the hard-copy plot and a call to Picture 1. Write Fortran subroutines VRELM and VRELD to MAIN to enable relative move and draw primatives to be executed. Write Fortran subroutine VSYMBL to enable symbols to be plotted on Versatec plotter in selected rotation and size.

Method:

- Write Fortran code to draw centred eight inch box
- Write Fortran code to call picture 1. (CALL PIC1)
- Write Fortran subroutine VRELM
- Write Fortran subroutine VRELD
- Write Fortran subroutine VSYMBL

SUBROUTINES CALLED:
TRMPLT

CALLED BY:
PASS2

## 5.12 SUBROUTINE DOTOOP(LINE,IOP)

Purpose: Process the op-codes and arguments of the current instruction which was read in from PASS1.DAT

Method:

- LINE is an eighteen character array read in from PASS1.DAT. It contains three fields, each of which may or may not be blank.
```
LINE(1)  - LINE(7)  is the LABEL field
LINE(8)  - LINE(11) is the OP-CODE field
LINE(12) - LINE(18) is the ARGUMENT field
```
- The instruction is identified by the value of IOP
- The arguments are decoded from the ARGUMENT field as appropriate

SUBROUTINES CALLED:
Nil

CALLED BY:
PASS2

## 5.13 SUBROUTINE VRSOUT(IOP)

Purpose: Write the translated Fortran instruction to PASS2.DAT

Method:

- The instruction and its argument/s have been dissassembled by subroutine DOTOOP and are ready for output.
- The instruction is recognized by the value of IOP
- The completed instruction is written to PASS2.DAT

SUBROUTINES CALLED:
Nil

CALLED BY:
PASS2

## 5.14 SUBROUTINE GETLIN(LINE,EOF)

Purpose: Read one line of data from the file PASS1.DAT

Method:

- Read the fields of the current line of PASS1.DAT into the array called LINE which has been declared BYTE LINE(18).

- Set the End-of-File flag TRUE if the end of PASS1.DAT was encountered.

SUBROUTINES CALLED:
Nil

CALLED BY:
PASS2

## 5.15 SUBROUTINE WHATOP(LINE,IOP)

Purpose: *Identify the Op-code of the instruction currently in the* array LINE.

Method:

- Encode elements 8 to 11 inclusive of LINE onto the real variable OCODE.

- Search through the array OPCODE (which contains a list of all Op-codes) for a match with OCODE.

- Set IOP to the integer matching that identified Op-code.

- If the Op-code is not recognized then determine if this line is a label by searching for a colon in one of the first seven elements of *LINE*.

- Set IOP = 20 if a label was recognized.

- If neither an Op-code nor a label was recognized then set IOP = 0

SUBROUTINES CALLED:
Nil

CALLED BY:
PASS2

## 5.16 SUBROUTINE TRMPLT

Purpose: Write a line of code to the file PASS2.DAT which will terminate the Versatec printer/plotter.

Method:

- Write the code 'CALL PLOT(0.0,0.0,999)' to the file PASS2.DAT

SUBROUTINES CALLED:
Nil

CALLED BY:
SETUPV

## REFERENCES

1. SANDERS Associates Inc.
Computer Graphics Display System.
Graphic Control Program Enhanced (GCP+)
Programmer's Reference Manual.
February 1980.

2. SANDERS Associates Inc.
Computer Graphics Display System.
Model 5753 2-D/3-D Coordinate Converter
User's Manual.
May 1980.

3. SANDERS Associates Inc.
Graphic-7
Fortran Support Package(FSP)
User's Manual
September 1980

4. Versatec Inc.
Versaplot Software Manual.
Publication No. 5721-03, Edition No. 3
September 1982

5. N.F. Hooke
Development of Computer Graphics Software
for the Display of Aircraft Combat Data in
Simulated Real-Time.
ARL TM to be published.

## DISTRIBUTION

AUSTRALIA

DEPARTMENT OF DEFENCE

CENTRAL OFFICE
Chief Defence Scientist )
Deputy Chief Defence Scientist ) (1 copy)
Superintendent, Science and Program Administration )
Controller, External Relations, Projects and Analytical Studies )
Defence Science Adviser (UK) (Doc Data sheet only)
Counsellor, Defence Science (USA) (Doc Data sheet only)
Defence Science Representative (Bangkok)
Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)

AERONAUTICAL RESEARCH LABORATORIES
Director
Library
Superintendent - Systems
Divisional File - Systems
Author: R Bateman
D A Bird

MATERIALS RESEARCH LABORATORIES
Director/Library

DEFENCE RESEARCH CENTRE
Library

AIR FORCE OFFICE
Air Force Scientific Adviser

CENTRAL STUDIES ESTABLISHMENT
Information Centre

GOVERNMENT AIRCRAFT FACTORIES
Manager
Library

SPARES (5 copies)

TOTAL (44 copies)

Department of Defence

# DOCUMENT CONTROL DATA

| | | | |
|---|---|---|---|
| 1. a. AR No<br>AR-003-989 | 1. b. Establishment No<br>ARL-SYS-TM-76 | 2. Document Date<br>JANUARY 1985 | 3. Task No<br>DST 84/127 |
| 4. Title<br>Program DISAS - A Computer Program to obtain Hard-Copy Plots of an Image Displayed on a Vector Graphics Device in a Local-Area-Network. | 5. Security<br>a. document<br>UNCLASSIFIED<br>b. title c. abstract<br>U U | 6. No Pages<br>20<br>7. No Refs<br>5 | |

8. Author(s)

Rodney BATEMAN

9. Downgrading Instructions

-

10. Corporate Author and Address

Aeronautical Research Laboratories
P.O. Box 4331
MELBOURNE, VIC. 3001

11. Authority (as appropriate)
a. Sponsor b. Security c. Downgrading d. Approval

-

12. Secondary Distribution (of this document)

Approved for Public Release

Overseas enquirers outside stated limitations should be referred through ASDIS, Defence Information Services Branch, Department of Defence, Campbell Park, CANBERRA ACT 2601

13. a. This document may be ANNOUNCED in catalogues and awareness services available to ...

No Limitation

13. b. Citation for other purposes (ie casual announcement) may be (select) unrestricted (or) as for 13 a.

14. Descriptors

Computer graphics
Computer networks
Applications program [computers]
Plotting

15. COSATI Group

09020

16. Abstract

Program DISAS is a program which supports the computer graphics facilities of Combat Effectiveness Group at ARL. It operates in a local-area-network incorporating a network of PDP11 minicomputers, a Sanders Graphic-7 vector graphics device and a Versatec electrostatic printer/plotter.

Program DISAS produces hard-copy plots on the Versatec printer/plotter of a selected image displayed on the screen of the SANDERS Graphic-7 vector graphics device.

A listing of the program can be obtained from the Principal Officer of Combat Effectiveness Group.

This page is to be used to record information which is required by the Establishment for its own use but which will not be added to the DISTIS data base unless specifically requested.

16. Abstract (Contd)

17. Imprint

Aeronautical Research Laboratories, Melbourne

18. Document Series and Number

Systems
Technical Memorandum 76

19. Cost Code

766160

20. Type of Report and Period Covered

-

21. Computer Programs Used

DISAS [Applications Program]

22. Establishment File Ref(s)

END

DATE
FILMED

6 - 85